Fitting and Tracking in 2-D and 3-D Images Using Wavelet Based Deformation Model

A Thesis Submitted
In Partial Fulfillment of the Requirements
For the Degree of

Master of Technology

By

Soma Biswas

to the

Department of Electrical Engineering

Indian Institute of Technology, Kanpur

May 2004

Certificate

This is to certify that the work contained in the thesis entitled "*Fitting and Tracking in 2-D and 3-D Images Using Wavelet Based Deformation Model*" by Soma Biswas has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

G Sharma

May 2004

(Dr. Govind Sharma)
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Abstract

This work concentrates on fitting and tracking of 2-D and 3-D images using a deformation model. The deformation model uses wavelets and also utilizes the interrelation between wavelets and function spaces (in this case Sobolev space) in order to impart a certain degree of smoothness to the fitted curves. The formulation defines a probabilistic model that induces a prior distribution for contour deformation. To increase the robustness of the approach, the wavelet models are expressed in terms of shape spaces. Based on this distribution, the fitting problem is solved in Bayesian terms. Several examples illustrating the success of the model in both 2-D and 3-D have been included. Also the same framework has been used for the tracking purpose. The deformation model is used to generate a prior dynamic model for contour evolution in time. This probabilistic model is then applied to solve the tracking problem. The tracking problems for both the 2-D and the 3-D cases have been solved using the Kalman filter.

Acknowledgement

I would like to express my profound gratitude to Prof. Govind Sharma for his constant support and guidance throughout this work. The frequent interactions with him helped me in solving the various problems, which I faced during my thesis. He has also been a source of inspiration and motivation without which it would not have been possible to complete this thesis.

I would also like to thank my friends Sangeeta, Kaustav, Tushar and Krishnendu who have made the stay at Kanpur very pleasant and memorable.

Last, but not the least, I am thankful to my parents and my sister for their support and encouragement.

Contents

1 **Introduction** 5

1.1 Some Previously used methods 6

1.1.1 Snakes 6

1.1.2 B-spline 7

1.1.3 Fourier Descriptors 7

1.2 Our Methodology 8

1.3 Organization of the thesis 8

2 **Background on Wavelets, Sobolev Space, Bayesian Estimation and Shape Space** **10**

2.1 Introduction 10

2.2 Wavelet Theory 10

2.2.1 Wavelet Series Expansion 12

2.2.2 Salient Features of Wavelet Transform 12

2.3 Wavelet Basis Functions 13

2.4 Sobolev Space 14

2.5 Bayesian Estimation 14

2.6 Shape Space 15

3 **Curve Fitting in 2-dimensions** **20**

3.1 Introduction 20

3.2 Wavelet Shape Representation 20

3.3 Wavelet Probabilistic Deformation Model In Sobolev Spaces 22

3.4 Wavelet Based Deformation Model In Shape Spaces 24

3.5 Probabilistic Model In Shape Spaces 25

3.6 The Fitting Problem 25

4 Curve Tracking **27**

4.1 Introduction 27

4.2 Theory of Kalman Filter 27

4.2.1 Kalman Filter Algorithm 29

4.3 Dynamical Model 31

5 Fitting and Tracking in 3-Dimensions **32**

5.1 Introduction 32

5.2 Wavelet Shape Representation in 3-Dimensions 32

5.3 Wavelet Probabilistic Deformation Model in 3-Dimensions 34

5.4 Shape Space for 3-Dimensions 35

5.5 Probabilistic Model In Shape Spaces 36

5.6 The Fitting Problem in 3-Dimensions 37

6 Implementation and Results **38**

6.1 Introduction 38

6.2 Algorithm for 2-D Curve fitting and Tracking 38

6.3 Algorithm for 3-D Curve fitting and Tracking 41

6.4 Simulation Results For 2-D Fitting 44

6.5 Simulation Results For 2-D Tracking 46

6.6 Simulation Results For 3-D Fitting 47

6.7 Simulation Results For 3-D Tracking 51

7 Conclusion and Future Scope **52**

7.1 Future Scope 53

List of Figures

2.1 (a) Haar scaling function, (b) Haar wavelet function 13

2.2 Control points undergoing arbitrary deformation 16

2.3 Six degrees of freedom of 2D affine transformation: translation vertically and horizontally, rotation and scaling vertically, horizontally and diagonally 18

4.1 Block Diagram For Kalman Filter Algorithm 28

6.1 Hand drawn prior around the heart. Arrows show the direction of searching from the hand drawn prior 39

6.2 (a) Noisy synthetic image, (b) Hand drawn prior provided around the object of interest, (c) Measurement data obtained from the noisy image, (d) Fitted curve around the object of interest 44

6.3 (a) Real image of a table tennis ball, (b) Hand drawn prior provided around the ball, (c) Measurement data (d) Fitted curve 45

6.4 (a) Image of a human heart, (b) Hand drawn prior provided around the heart, (c) Measurement data (d) Fitted curve 45

6.5 Rowwise from top : Some Frames from the tracking results in a noisy synthetic image 46

6.6 Rowwise from top : Some Frames from the tracking results of a table tennis ball in a real image 46

6.7 3-D fitting results on MRI slices of human brain 47

6.8 3-D fitting results on MRI slices of human brain: contd. 48

6.9 (a) Hand drawn prior for slice 1, (b) Hand drawn prior for slice 5, (c) Measurement for slice 1, (d) Measurement for slice 5 49

6.10 3D rendering for the fitted MRI data 50

6.11 (a) 3D rendering of synthetic 3-D data, (b) Slice data for the above image with the fitted data shown as wire frame, (c) 3D rendering of fitted data 50

6.12 2 frames from the 3D synthetic image used for tracking 51

6.13 Tracked movement of the individual slices 51

Chapter 1

Introduction

Boundary representation is very important in shape description and recognition [1]. It plays a key role in many applications such as image analysis, pattern recognition, computer graphics and computer-aided animation. In many cases, such as in medical images, finding the boundary of structures, segmenting and reconstructing a compact geometric representation of these structures is difficult due to the huge number of data sets and the complexity and variability of the shapes of interest. Furthermore, the shortcomings of sampled data, such as sampling artifacts, spatial aliasing and noise, may cause the boundaries of structures to be indistinct and disconnected. Traditional boundary finding techniques which consider only local information usually do not give satisfactory results as the edges that are found do not necessarily correspond to the boundaries of the objects. With the exception of high-quality images from controlled environments, they produce spurious edges and gaps and so they are of limited use in general and of no use in poor quality images. As a result, these model-free techniques usually require considerable amounts of expert intervention.

Among model-based techniques, deformable models offer a unique and powerful approach to image analysis. Deformable models are object models that possess shape varying capability, which makes them suitable for representing non-rigid objects. They have been found to be effective in segmenting, matching and tracking different structures by exploiting constraints derived from the image data together with the *a priori* knowledge about the location, size and shape of these structures. Apart from computer vision and computer graphics, they have found wide application in medical image analysis. Among other applications, 2-dimensional and 3-dimensional deformable models have been used to fit and track a variety of structures widely ranging in scale and variety. For example, they are used to fit and subsequently track the non-rigid motion of the heart.

Deformable model geometry usually permits broad shape coverage by employing geometric representation that involve many degrees of freedom. The model remains manageable, however, because the degrees of freedom are generally not permitted to evolve independently. Usually deformations are confined in a region of the whole space and prior information about preferred deformations of the model can be obtained. This preference is then expressed in terms of a deformation energy (or cost) that penalizes some deformations. The importance of this prior information is that it simplifies and increases the robustness of the solutions to some problems like fitting and tracking. The deformation energy is coupled with a data mismatch criterion that measures the degree of discrepancy between the model and some measures extracted from an image. Model matching can then be formulated as an optimization problem of a combined criterion function that is defined in terms of both the deformation and the mismatch energy.

An important subset of deformable models is the set of active contour models where an object is characterized by its external shape. Some commonly used deformable models are discussed in the following section.

1.1 Some Previously used methods

1.1.1 Snakes

Snakes [2] are planar deformable contours that are often used to approximate locations and shapes of object boundaries in images based on the reasonable assumption that boundaries are piecewise continuous or smooth. Geometrically, a snake is a parametric contour embedded in the image plane. The shape of the contour is dictated by a functional which represents the energy of the contour and the final shape of the contour corresponds to the minimum of this energy. The functional consists of two terms. The first term represents the internal deformation energy which characterizes the deformation of a stretchy, flexible contour. The two physical parameter functions that dictate the physical characteristic of the contour are the tension of the contour and the rigidity. The second term couples the snake to the image. It is a scalar potential defined on the image plane.

1.1.2 B-spline

Another way to represent image contours is to use piecewise polynomials. This representation is compact, local (i.e. a small change in the original curve does not effect the entire representation) and also it allows continuous approximation, even over gaps in the data. The approach commonly used consists of first extracting a set of knots from the discrete curve and then approximating the elementary curves between each pair of knots by a B-spline [3]. A B-spline is a piecewise polynomial, which is expressed as a linear combination of basis functions, which are themselves piecewise polynomials where the coefficients are the vertices of the B-spline guiding polygon. They can be manipulated by modifying its guiding polygon and as they are defined locally, modifying the position of a vertex only affects the B-spline locally.

1.1.3 Fourier Descriptors

In some research work [4], Fourier descriptors have been used to represent contours that are smooth and continuously deformable. The parameters of the deformable contours are the Fourier coefficients. A probability distribution on the Fourier coefficients is specified so that there is a flexible bias towards some particular shape. The spread of the distribution is governed by the variability of the object class. A Bayesian decision rule is then used to obtain the optimal fitting of the boundary to the image.

1.2 Our Methodology

For this thesis, the relation between wavelets and smoothness spaces (namely Sobolev Space) have been used to build wavelet contour representations [5] with a desired degree of smoothness. Wavelet descriptors are very efficient in representing and detecting local features of a curve due to spatial and frequency localization property of wavelet bases. For this type of representation, different levels of smoothness can be defined and smooth deformations can be enforced without the necessity to alter the balance between the uncertainty of the prior model for deformations and data extracted from the image. This wavelet representation is set in probabilistic terms so that the realizations of the stochastic model are almost sure in a predefined smoothness space. To increase the robustness of the approach, the wavelet models are expressed in terms of shape spaces. Then Bayesian estimation has been used to find the optimum fitted contour.

To solve the tracking problem, Kalman filter has been used. The tracking problem is a two-phase problem in which a dynamic model taken as an autoregressive process is used for prediction from one time step to the next. Then the predicted contour is refined using the measures obtained from the image and these two phases are repeated again with the next frame from the sequence. Both the fitting and tracking problem has been implemented successfully in both 2-dimensions and 3-dimensions.

1.3 Organization of the thesis

In Chapter 2 of the thesis, some background material is provided which includes some portions of wavelet transform and their salient features, Sobolev Space, inter-relation between Sobolev Space and wavelet transform coefficients, Bayesian estimation and Shape space. The theory of curve fitting in 2-dimensions has been discussed in details in Chapter 3. Chapter 4 of the thesis deals with Kalman filter theory which has been used for tracking in both 2-D and 3-D and also the dynamical model which has been considered. The theory of surface fitting and tracking in 3-D is covered in Chapter 5. Implementation algorithms for fitting and tracking for both 2-D and 3-D cases are given in Chapter 6 along with the implementation results for both cases. The thesis ends with the conclusion and scope for future research in Chapter 7.

Chapter 2

Background on Wavelets, Sobolev Space, Bayesian Estimation and Shape Space

2.1 Introduction

In this chapter, some background material is presented which will be required for the understanding of the thesis. First, a brief overview of the wavelet theory [6][7][8] and some salient features of wavelet transform are given. The theory of Sobolev space [5][9], basics of Bayesian estimation [10] and Shape space [5][11][12] are then discussed.

2.2 Wavelet Theory

Multiresolution theory is concerned with the representation and analysis of signals or images at more than one resolution. So features which might go undetected at one resolution may be easy to spot at another. In multiresolution analysis (MRA), a scaling function is used to create a series of approximations of a function or image. Additional functions, called wavelets are then used to encode the difference in information between adjacent approximations.

A function *f(x)* can be expressed as a linear combination of expansion functions

$$f(x) = \sum_k \alpha_k \varphi_k(x) \tag{2.1}$$

where k is an integer index of the finite or infinite sum, α_k are real valued *expansion coefficients* and $\varphi_k(x)$ are real valued *expansion functions.*

Consider the expansion functions are composed of integer translations and binary scalings of the real square-integrable function $\varphi(x)$ i.e. the set $\{\varphi_{j,k}(x)\}$, where

$$\varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k) \tag{2.2}$$

for all $j, k \in \mathbf{Z}$ and $\varphi(x) \in L^2(\mathbf{R})$. Here k determines the position of $\varphi_{j,k}(x)$ along the x-axis, j determines its width and $2^{j/2}$ controls its amplitude. Because the shape of $\varphi_{j,k}(x)$ changes with j, $\varphi(x)$ is called a *scaling function.* By choosing $\varphi(x)$ properly, $\{\varphi_{j,k}(x)\}$ can be made to span $L^2(\mathbf{R})$, the set of all measurable, square-integrable functions.

The subspace spanned over k for any j is denoted as

$$V_j = \overline{\mathrm{Span}_k\{\varphi_{j,k}(x)\}} \tag{2.3}$$

The scaling function obeys the four fundamental requirements of MRA.

- The scaling function is orthogonal to its integer translates.
- The subspaces spanned by the scaling function at low scales are nested within those spanned at higher scales.

$$V_{-\infty} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \subset V_\infty \tag{2.4}$$

- The only function that is common to all V_j is $f(x) = 0$
- Any function can be represented with arbitrary precision.

Given a scaling function, we can define a *wavelet function* $\psi(x)$, that together with its integer translates and binary scalings, spans the difference between any two adjacent scaling subspaces. It is defined as

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \qquad \forall k \in \mathbf{Z} \tag{2.5}$$

The subspace spanned over k for any j is denoted as

$$W_j = \overline{\mathrm{Span}_k\{\psi_{j,k}(x)\}} \tag{2.6}$$

The scaling and wavelet function subspaces are related by

$$V_{j+1} = V_j \oplus W_j \tag{2.7}$$

where $\oplus$ denotes the union of spaces.

The space of all measurable, square-integrable functions can now be expressed as

$$L^2(\mathbf{R}) = V_{j_0} \oplus W_{j_0}' \oplus W_{j_0+1} \oplus \ldots \tag{2.8}$$

2.2.1 Wavelet Series Expansion

The wavelet series expansion of a function $f(x) \in L^2(\mathbf{R})$ is given by

$$f(x) = \sum_{l} c_{j_0,l}\varphi_{j_0,l}(x) + \sum_{j=j_0}^{\infty}\sum_{l} d_{j,l}\psi_{j,l}(x) \tag{2.9}$$

where j_0 is an arbitrary starting scale. The $c_{j_0,l}$ are called *approximation* or *scaling coefficients* and $d_{j,l}$ are called *detail* or *wavelet coefficients*. If the expansion functions form an orthonormal basis then the expansion coefficients can be calculated as

$$c_{j_0,l} = \langle f(x), \varphi_{j_0,l}(x)\rangle = \int f(x)\varphi_{j_0,l}(x)dx \tag{2.10}$$

and

$$d_{j,l} = \langle f(x), \psi_{j,l}(x)\rangle = \int f(x)\psi_{j,l}(x)dx \tag{2.11}$$

2.2.2 Salient Features of Wavelet Transform

There are a number of salient features in wavelet transforms that make wavelet-domain processing attractive.

- Locality: Each wavelet coefficient represents the signal content localized in spatial location and frequency.
- Multiresolution: The wavelet transform analyses the signal at a nested set of scales with different resolutions.

- Energy compaction: A wavelet coefficient is large only if singularities are present within the support of the wavelet basis. Therefore in contrast with other approaches, we need to model only a small number of coefficients. This is of great importance in real-time applications.

2.3 Wavelet Basis Functions

Some commonly used wavelet basis functions are *Haar wavelet*, *Shannon wavelet*, *Daubechies wavelet*, *Biorthogonal wavelet*, *Butterworth wavelet*, etc. For this thesis, Haar basis function has been used.

Haar Basis Function The Haar scaling function is defined as

$$\varphi(t) = \begin{cases} 1 & \text{for } 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$$

and Haar wavelet is given by

$$\psi(t) = \begin{cases} 1 & \text{for } 0 \leq t < \frac{1}{2} \\ -1 & \text{for } \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$$

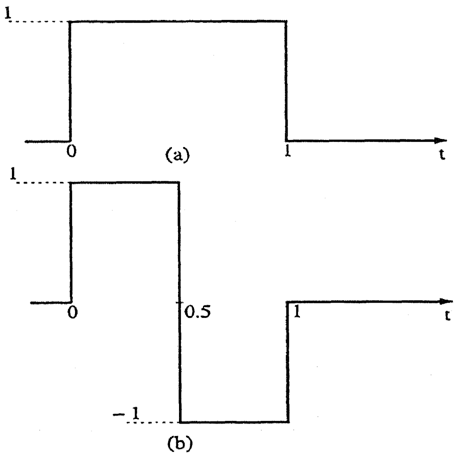


Figure 2.1: (a) Haar scaling function, (b) Haar wavelet function

2.4 Sobolev Space

A function space is a vector space whose points are functions. Sobolev Space is one type of function space. The space L^p is the set of measurable functions f which are p-integrable. A function $f(x)$ is said to be p-integrable if

$$\int_{-\infty}^{\infty} \mid f(x)|^p dx \tag{2.12}$$

is finite.

The Sobolev space $L^{p,s}(\mathbb{R}) \equiv L^{p,s}$ for $1 < p < \infty$ and $s = 1, 2, 3, \ldots$ is defined to be the space of all functions $f \in L^p(\mathbb{R})$ such that, for all $n = 1, 2, \ldots, s$, the n^{th} derivative of f also belongs to $L^p(\mathbb{R})$. The following quantity is the norm on the space $L^{p,s}$.

$$\| f\|_{L^{p,s}} = \| f\|_{L^p} + \sum_{n=1}^{s} \| D^n f\|_{L^p} \tag{2.13}$$

where $D^n f$ is the n^{th} derivative of the function f.

The Sobolev norm of a function is related to its smoothness. Again, this norm can be expressed in terms of the scaling and wavelet coefficients of the function. Thus the smoothness of the function is related to its scaling and wavelet coefficients. A curve belongs to the Sobolev space if its co-ordinate functions belong to that space. Like in the case of functions, for curves also the smoothness of the contour is related to the coefficients of its representation by means of the contour norm in Sobolev space.

2.5 Bayesian Estimation

Suppose that the distribution of a RV (random variable) $\mathbf{x}$ is a function $F(x, \theta)$ of known form depending on a parameter θ and we wish to estimate θ. To do so, the underlying physical experiment is repeated n times and the observed values of $\mathbf{x}$ are denoted by x_i. If θ is viewed as an unknown constant and the estimate is solely based on the observed values x_i of the RV $\mathbf{x}$, then it is called *classical estimation.* But in certain applications, θ is not totally unknown. In bayesian statistics, the available prior information about θ is used in the estimation problem. In this approach, the unknown parameter θ is viewed as the value of an RV $\boldsymbol{\theta}$ and the distribution of $\mathbf{x}$ is interpreted as the conditional distribution $F_x(x \mid \theta)$ of $\mathbf{x}$ assuming $\boldsymbol{\theta} = \theta$. The prior information is used to assign a density $f_\theta(\theta)$ to the RV $\boldsymbol{\theta}$, and the problem is to estimate the value θ in terms of the observed values x_i of $\mathbf{x}$ and the density of $\boldsymbol{\theta}$.

To solve a bayesian estimation problem, we first assume that no observations are available. The available information is the *prior* density $f_\theta(\theta)$ of $\boldsymbol{\theta}$ and the problem now is to find a constant $\hat{\theta}$ close in some sense to the unknown θ. If LMS criterion is used for selecting $\hat{\theta}$, then

$$\hat{\theta} = E\{\boldsymbol{\theta}\} = \int_{-\infty}^{\infty} \theta f_\theta(\theta) d\theta \tag{2.14}$$

Now if the observations from the experiments are incorporated, and the same criterion is used for estimation

$$\hat{\theta} = E\{\boldsymbol{\theta} \mid X\} = \int_{-\infty}^{\infty} \theta f_\theta(\theta \mid X) d\theta \tag{2.15}$$

where $X = [x_1, \ldots, x_n]$ and

$$f_\theta(\theta \mid X) = \frac{f(X \mid \theta)}{f(X)} f_\theta(\theta) \tag{2.16}$$

Here $f(X \mid \theta)$ is the conditional density of the n RV x_i assuming $\boldsymbol{\theta} = \theta$. The density $f_\theta(\theta)$ is called *prior* (prior to the measurements), the density $f_\theta(\theta \mid X)$ is called *posterior* (after the measurements) and the density $f(X \mid \theta)$ is called the *likelihood function.*

2.6 Shape Space

Let $\mathbf{r}(u) = (x(u), y(u))$ be a parameterized closed planar curve that represents the shape of our object of interest. The parameterization in terms of B-splines or wavelet coefficients can be expressed as

$$\mathbf{r}_u = (\mathbf{G}(u)^T \mathbf{w}^x, \mathbf{G}(u)^T \mathbf{w}^y) \qquad 0 \leq u \leq L \tag{2.17}$$

where $\mathbf{G}(u)^T$ is the vector of B-spline or wavelet basis functions, $\mathbf{w}^x$ and $\mathbf{w}^y$ are vectors of B-spline control point coordinates or scaling and wavelet coefficients and L is the span of the parameter u. The B-spline control points or the wavelet coefficients can be combined in a spline vector or a wavelet vector as

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}^x \\ \mathbf{w}^y \end{pmatrix} \tag{2.18}$$

In certain applications, it is necessary to reduce the number of degrees of freedom in the model describing the shape of the deformable object. For example, taking $\mathbf{w}$ to be a spline-vector, from Figure 2.2 it is clear that if all the possible degrees of freedom of the spline-vector are manipulated arbitrarily, then many uninteresting shapes are generated which are not at all reminiscent of faces. Restricting the displacements of control points is more meaningful as it preserves the face-like quality of the shape. Shape space is a parameterization of the set of allowed deformations of a base curve.

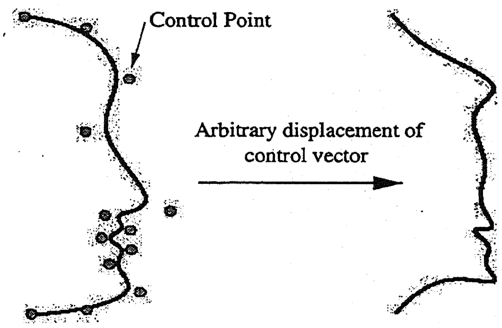


Figure 2.2: Control points undergoing arbitrary deformation

A linear shape space $\mathcal{M} = \mathcal{L}(H, \bar{\mathbf{w}})$ is a linear mapping of a *shape space vector* $\mathbf{x} \in \mathbb{R}^{N_X}$ to the vector $\mathbf{w} \in \mathbb{R}^{N_W}$:

$$\mathbf{w} = H\mathbf{x} + \bar{\mathbf{w}} \tag{2.19}$$

where H is a $N_W \times N_X$ *shape matrix*. The vector $\bar{\mathbf{w}}$ represents a *base* or *template curve* against which shape variations are measured. For instance, a class of shapes consisting of $\bar{\mathbf{w}}$ and curves close to $\bar{\mathbf{w}}$ could be expressed by restricting the shape space $\mathcal{M}$ to small $\mathbf{x}$. There can be several types of shape spaces some of which are explained below.

1. **Planar Affine Shape Space :** For a planar shape, just six affine degrees of freedom are required to describe to a good approximation, the possible shapes of its bounding curve. The six possible deformations are illustrated in Figure 2.3. The planar affine group can be viewed as a class of all linear transformations that can be applied to a template curve $\mathbf{r}_0(u)$:

$$\mathbf{r}(u) = \mathbf{t} + M\mathbf{r}_0(u) \tag{2.20}$$

where $\mathbf{t} = (t_1, t_2)^T$ is a two-dimensional translational vector and M is a 2×2 matrix, so that $\mathbf{t}$ and M together represent the 6 degrees of freedom of the space. This class can be represented as a shape space with template $\bar{\mathbf{w}}$ and shape matrix

$$H = \begin{pmatrix} \mathbf{T} & \mathbf{0} & \bar{\mathbf{w}}^x & \mathbf{0} & \mathbf{0} & \bar{\mathbf{w}}^y \\ \mathbf{0} & \mathbf{T} & \mathbf{0} & \bar{\mathbf{w}}^y & \bar{\mathbf{w}}^x & \mathbf{0} \end{pmatrix} \tag{2.21}$$

where $\mathbf{T} = (11\ldots1)^T$ or $\mathbf{T} = (10\ldots0)^T$ depending upon whether B-splines or wavelets have been used and $\mathbf{0} = (00\ldots0)^T$ are vectors each with N components where N is the dimension of $\bar{\mathbf{w}}^x$ or $\bar{\mathbf{w}}^y$. The first two columns of H represent horizontal and vertical translation. The remaining four affine motions do not correspond one-for-one to the last four columns of H. But they can be expressed as linear combinations of these columns. From the shape space transformation in equation (2.19), it is clear that the elements of $\mathbf{x}$ act as weights on the columns of H. The interpretation of those weights in terms of planar transformations of the template is:

$$\mathbf{x} = (t_1, t_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})^T \tag{2.22}$$

Some examples of transformations are:

- $\mathbf{x} = (0,0,0,0,0,0)^T$ represents the original template shape $\bar{\mathbf{w}}$
- $\mathbf{x} = (1,0,0,0,0,0)^T$ represents the template translated 1 unit to the right
- $\mathbf{x} = (0,0,1,1,0,0)^T$ represents the template doubled in size
- $\mathbf{x} = (0,0,\cos\theta - 1, \cos\theta - 1, -\sin\theta, \sin\theta)^T$ represents the template rotated through angle θ
- $\mathbf{x} = (0,0,1,0,0,0)^T$ represents the template doubled in width

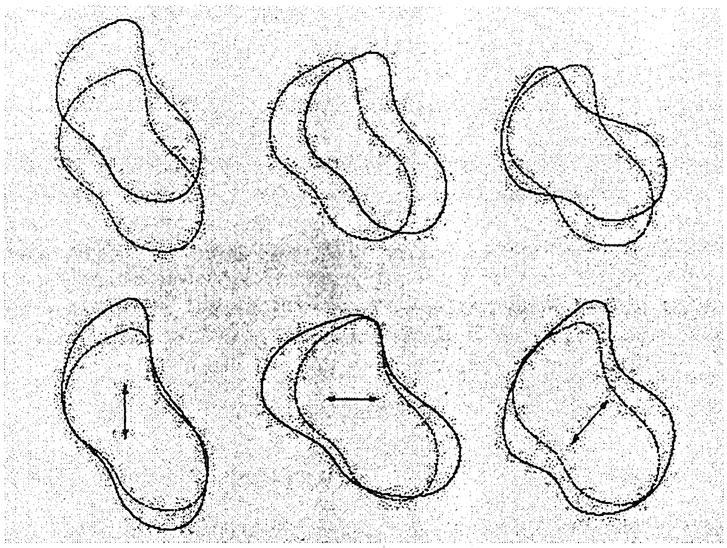


Figure 2.3: Six degrees of freedom of 2D affine transformation: translation vertically and horizontally, rotation and scaling vertically, horizontally and diagonally

2. **Euclidean similarity shape space:** The similarity space for a template curve $\mathbf{r}_0(s)$ with parameter vector $\bar{\mathbf{w}}$ form a four-dimensional shape space with *shape matrix*

$$H = \begin{pmatrix} \mathbf{T} & \mathbf{0} & \bar{\mathbf{w}}^x & -\bar{\mathbf{w}}^y \\ \mathbf{0} & \mathbf{T} & \bar{\mathbf{w}}^y & \bar{\mathbf{w}}^x \end{pmatrix} \tag{2.23}$$

The first 2 columns are associated with translations while the rest are associated with dilation and rotation.

With this definition for the shape matrix, the *shape space vector*

$$\mathbf{x} = (t_1, t_2, k(\cos(\theta) - 1), k(\sin(\theta))) \tag{2.24}$$

translates the curve determined by $\bar{\mathbf{w}}$ t_1 units on the x-axis and t_2 units on the y-axis, obtaining also a rotation with angle θ and a scaling with factor k.

Apart from these there are some other shape spaces like *Key Frames Shape Space*[5].

Chapter 3

Curve Fitting in 2-dimensions

3.1 Introduction

This chapter deals with the theory of curve fitting in 2-dimensions [5]. First, the representation of curves in terms of its wavelet coefficients is given. Then a probabilistic model of the curve is formulated which is followed by the formulation of wavelet based deformation model in shape space. Finally, the fitting problem is solved using Bayesian estimation.

3.2 Wavelet Shape Representation

Let $\mathbf{r}(u) = (x(u), y(u))$ be a parameterized closed planar curve that represents the shape of an object of interest. If the wavelet transform is applied independently to each of the $x(u)$, $y(u)$ functions, we can describe the planar curve in terms of the decomposition of $\mathbf{r}(u)$:

$$\mathbf{r}(u) = \sum_{l} \mathbf{c}_{jo,l}\varphi_{jo,l}(u) + \sum_{j=jo}^{\infty}\sum_{l} \mathbf{d}_{j,l}\psi_{j,l}(u) \tag{3.1}$$

where

$$\mathbf{c}_{jo,l} = (c_{jo,l(x)}, c_{jo,l(y)})^T \tag{3.2}$$

$$\mathbf{d}_{j,l} = (d_{j,l(x)}, d_{j,l(y)})^T \tag{3.3}$$

Here j_0 is an arbitrary starting scale. The $c_{j_0,l(x)}$ and $d_{j,l(x)}$ are the *scaling coefficients* and *wavelet coefficients* for the function $x(u)$ and $c_{j_0,l(y)}$ and $d_{j,l(y)}$ are the *scaling coefficients* and *wavelet coefficients* for the function $y(u)$. $\mathbf{c}_{j_0,l}$ and $\mathbf{d}_{j,l}$ are the combined *scaling coefficients* and *wavelet coefficients* for the curve $\mathbf{r}(u)$. The parameter is denoted by u and T denotes the transpose operator.

If the curve $\mathbf{r}(u)$ is closed and the parameter u belongs to an interval $I = [0, L]$, then we obtain the representation :

$$\mathbf{r}(u) = \mathbf{c}_{0,0}\varphi_{0,0}(u) + \sum_{j=0}^{\infty} \sum_{0 \leq l < 2^j} \mathbf{d}_{j,l}\psi_{j,l}(u) \tag{3.4}$$

Here the starting scale is taken as 0. The coefficients in the curve expansion is defined as :

$$\mathbf{c}_{0,0} = (\langle x(u), \varphi_{0,0}(u)\rangle, \langle y(u), \varphi_{0,0}(u)\rangle)^T \tag{3.5}$$

$$\mathbf{d}_{j,l} = (\langle x(u), \psi_{j,l}(u)\rangle, \langle y(u), \psi_{j,l}(u)\rangle)^T \tag{3.6}$$

Here $\langle , \rangle$ denotes the scalar product :

$$\langle f, g\rangle = \frac{1}{L}\int_0^L f(u)g(u)du \tag{3.7}$$

The scaling and wavelet functions are normalized so that:

$$\| \varphi_{0,0}\|_{L_2(I)} = \| \psi_{j,l}\|_{L_2(I)} = 1 \qquad \text{with} \qquad \| f\|^2_{L_2(I)} = \langle f, f\rangle \tag{3.8}$$

In practice, only a finite number of coefficients are considered. So the representation becomes :

$$\mathbf{r}(u) = \mathbf{c}_{0,0}\varphi_{0,0}(u) + \sum_{j=0}^{J} \sum_{0 \leq l < 2^j} \mathbf{d}_{j,l}\psi_{j,l}(u) \tag{3.9}$$

Curve expansion can be written in concise form. First the scaling and wavelet functions are expressed in vector form as :

$$\mathbf{G}_W(u) = (\varphi_{0,0}(u), \psi_{0,0}(u), \ldots, \psi_{J-1,2^{J-1}-1}(u))^T \tag{3.10}$$

then the coordinate functions can be written as :

$$x(u) = \mathbf{G}_W(u)^T\mathbf{w}^x \qquad \text{and} \qquad y(u) = \mathbf{G}_W(u)^T\mathbf{w}^y \tag{3.11}$$

with scaling and wavelet coefficient vectors $\mathbf{w}^x$ and $\mathbf{w}^y$ defined as :

$$\mathbf{w}^x = \left(c_{0,0(x)}, d_{0,0(x)}, \ldots, d_{J-1,2^{J-1}-1(x)}\right)^T \tag{3.12}$$

$$\mathbf{w}^y = \left(c_{0,0(y)}, d_{0,0(y)}, \ldots, d_{J-1,2^{J-1}-1(y)}\right)^T \tag{3.13}$$

To describe the curve in matrix form, we define :

$$\mathbf{F}_W(u) = \mathbf{I}_2 \otimes \mathbf{G}_W(u)^T = \begin{pmatrix} \mathbf{G}_W(u)^T & \mathbf{0}^T{}_N \\ \mathbf{0}^T{}_N & \mathbf{G}_W(u)^T \end{pmatrix} \tag{3.14}$$

where $\otimes$ denotes Kronecker product and $\mathbf{0}_N$ a null vector of N components where $N = 2^J$.

If we define $\mathbf{w} = (\mathbf{w}^{xT}, \mathbf{w}^{yT})^T$, then we may write

$$\mathbf{r}(u) = \mathbf{F}_W(u)\mathbf{w} \tag{3.15}$$

3.3 Wavelet Probabilistic Deformation Model In Sobolev Spaces

The contour fitting and tracking problem can be formulated in probabilistic terms that allow an explicit representation for the uncertainty. The simplest wavelet transform statistical models are obtained by assuming that the coefficients are independent. Under the independence assumption, modeling reduces to simply specifying the marginal distribution of each scaling and wavelet coefficients. Here Gaussian distribution is employed. It can be shown that the Gaussian distribution is related to Sobolev spaces by means of the following result :

Theorem 3.3.1 *Let $f(x)$ be a real function where x is a real variable. Let it be decomposed in wavelet coefficients and suppose each coefficient is independently and identically distributed as*

$$d_{j,k} \sim N(0, \sigma^2{}_j), \qquad \sigma_j = 2^{-j\beta}\sigma_{DEF} \tag{3.16}$$

with $\beta > 0$ and $\sigma_{DEF} > 0$. Then the realizations of the probabilistic model are almost surely in the Sobolev space if and only if $\beta > s + 1/2$.

This result is extended to contours by the following theorem:

Theorem 3.3.2 *Let a curve* $\mathbf{C} \equiv \mathbf{r}(u) = (x(u), y(u))$ *be decomposed on its wavelet representation and suppose that the vectors of its representation* $\mathbf{d}_{j,k}$ *are independently and identically distributed as :*

$$\mathbf{d}_{j,k} = (d_{j,k(x)}, d_{j,k(y)})^T \sim N(\mathbf{0}_2, \sigma^2{}_j \mathbf{I}_2) \tag{3.17}$$

$$\sigma_j = 2^{-j\beta}\sigma_{DEF}, \qquad \beta > 0 \qquad and \qquad \sigma_{DEF} > 0 \tag{3.18}$$

then the realizations of the model are almost surely in the curve extension of the Sobolev space if and only if $\beta > s + 1/2$.

This result shows that the exponential decay of the variances $\sigma_j = 2^{-j\beta}\sigma_0$ can be used to enforce the smoothness of the contours generated from the probabilistic model. The greater the value of β, smaller is the variance and so the contours generated are smoother.

To complete the definition of the probabilistic model, the distribution of the scaling coefficient $\mathbf{c}_{0,0}$ is to be specified. This coefficient is associated with the translation of shape and it is assumed to be normally distributed and independent of the non-translation components $\mathbf{d}_{j,k}$.

$$\mathbf{c}_{0,0} = \begin{pmatrix} c_{0,0(x)} \\ c_{0,0(y)} \end{pmatrix} \sim N(\mathbf{0}_2, \sigma^2{}_{TR}\mathbf{I}_2) \tag{3.19}$$

Assuming these distributions, the wavelet probabilistic deformation model can be expressed in matrix form as

$$P(\mathbf{r}) \equiv P(\mathbf{w}) \propto exp(-\frac{1}{2}\mathbf{w}^T \mathrm{S}_w \mathbf{w}), \qquad \mathbf{r} = \mathbf{F}_W \mathbf{w} \tag{3.20}$$

Here S_w is the covariance matrix of the vector $\mathbf{w}$ where

$$\Sigma = \mathrm{S}_w{}^{-1} = \mathbf{I}_2 \otimes \begin{pmatrix} \sigma^2{}_{TR} & 0 & 0 & \dots & 0 \\ 0 & \sigma^2{}_{DEF}2^{-2\beta(0)} & 0 & \dots & 0 \\ 0 & 0 & \sigma^2{}_{DEF}2^{-2\beta(1)} & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & \sigma^2{}_{DEF}2^{-2\beta(J-1)} \end{pmatrix} \tag{3.21}$$

Here β is related to the smoothness of the deformation and $\sigma^2{}_{TR}$ and $\sigma^2{}_{DEF}$ are related to the uncertainty of the contour.

3.4 Wavelet Based Deformation Model In Shape Spaces

Shape Space has been discussed in details in Chapter 1. It is a parameterization of the set of allowed deformations of a base curve. A linear shape apace $\mathcal{M} = \mathcal{L}(H, \bar{\mathbf{w}})$ is a linear mapping of a *shape space vector* $\mathbf{x} \in \mathbb{R}^{N_X}$ to the vector $\mathbf{w} \in \mathbb{R}^{N_W}$:

$$\mathbf{w} = H\mathbf{x} + \bar{\mathbf{w}} \tag{3.22}$$

where H is a $N_W \times N_X$ *shape matrix.* The vector $\bar{\mathbf{w}}$ represents a base or template curve against which shape variations are measured. Here *planar affine shape space* has been used.

Planar Affine Shape Space : Here 6 degrees of freedom are allowed. The planar affine group can be viewed as a class of all linear transformations that can be applied to a template curve $\mathbf{r}_0(u)$:

$$\mathbf{r}(u) = \mathbf{t} + M\mathbf{r}_0(u) \tag{3.23}$$

where $\mathbf{t} = (t_1, t_2)^T$ is a two-dimensional translational vector and M is a 2 × 2 matrix, so that $\mathbf{t}$ and M together represent the 6 degrees of freedom of the space. This class can be represented as a shape space with template $\bar{\mathbf{w}}$ and shape matrix

$$H = \begin{pmatrix} \mathbf{T} & \mathbf{0} & \bar{\mathbf{w}}^x & \mathbf{0} & \mathbf{0} & \bar{\mathbf{w}}^y \\ \mathbf{0} & \mathbf{T} & \mathbf{0} & \bar{\mathbf{w}}^y & \bar{\mathbf{w}}^x & \mathbf{0} \end{pmatrix} \tag{3.24}$$

where $\mathbf{T} = (10\ldots0)^T$ and $\mathbf{0} = (00\ldots0)^T$ are vectors each with N components. The shape space vector is given by :

$$\mathbf{x} = (t_1, t_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})^T \tag{3.25}$$

3.5 Probabilistic Model In Shape Spaces

The probabilistic model in wavelet space induces a probabilistic model in shape space:

$$P(\mathrm{x}) \propto exp(-\frac{1}{2}\mathrm{x}^T \mathrm{S}_x \mathrm{x}) \tag{3.26}$$

where S_x is the covariance matrix of the vector x. Since $\mathrm{w} = H\mathrm{x} + \bar{\mathrm{w}}$, so we can write

$$\mathrm{x} = H^+(\mathrm{w} - \bar{\mathrm{w}}) \tag{3.27}$$

where H^+ is the pseudo-inverse of H. If H is a $m \times n$ matrix, then the pseudo-inverse H^+ is a $n \times m$ matrix defined as

$$H^+ = (H^T H)^{-1} H^T \tag{3.28}$$

Here since $\bar{\mathrm{w}}$ is the combined wavelet vector of the base curve, say $\bar{\mathrm{x}}$, so $\bar{\mathrm{x}} = 0$. This is also evident from the fact that shape space vectors are basically deviations from the base or template curve. Then the inverse of the covariance matrix S_x is given by:

$$\mathrm{S}_x{}^{-1} = H^+ \mathrm{S}_w{}^{-1} H^{+T} \tag{3.29}$$

3.6 The Fitting Problem

For this thesis, Bayesian estimation has been used to find the optimum shape space vector of the contour. The fitting problem can be defined in Bayesian terms as follows:

The *prior density* is given by the probabilistic model in shape space, i.e.

$$P(\mathrm{x}) \propto exp(-\frac{1}{2}\mathrm{x}^T \mathrm{S}_x \mathrm{x}) \tag{3.30}$$

From the image, some measurements of the contour x_D are obtained. The likelihood function is defined as :

$$P(\mathrm{r}_D \mid \mathrm{r}) = P(\mathrm{w}_D \mid \mathrm{w}) = P(\mathrm{x}_D \mid \mathrm{x}) \propto exp(-\frac{1}{2}(\mathrm{x} - \mathrm{x}_D)^T \mathrm{S}_D (\mathrm{x} - \mathrm{x}_D)) \tag{3.31}$$

with the matrix $\mathbf{S}_D$ given by:

$$\mathbf{S}_D{}^{-1} = H^{+}\sigma_D{}^{2}H^{+T} \tag{3.32}$$

Here σ_D is the variance of $\mathbf{w}_D - \bar{\mathbf{w}}$ where the wavelet and shape spaces are related by :

$$\mathbf{r}_D = \mathbf{F}_W\mathbf{w}_D \qquad \mathbf{w}_D = H\mathbf{x}_D + \bar{\mathbf{w}} \tag{3.33}$$

From Bayesian estimation, we get the maximum aposteriori estimate $\hat{\mathbf{x}}$ as:

$$\hat{\mathbf{x}} = (\mathbf{S}_x + \mathbf{S}_D)^{-1}\mathbf{S}_D\mathbf{x}_D \tag{3.34}$$

where the measurement vector $\mathbf{x}_D$ is given by $\mathbf{x}_D = H^{+}(\mathbf{w}_D - \bar{\mathbf{w}})$

Chapter 4

Curve Tracking

4.1 Introduction

In this chapter, the theory of Kalman Filter is discussed. Also, the dynamical model which has been used in the thesis for tracking problem is given.

4.2 Theory of Kalman Filter

A distinctive feature of the Kalman filter [13][14] is that its mathematical formulation is described in terms of state space concepts. Another novel feature of the Kalman filter is that its solution is computed recursively, applying without modification to stationary as well as nonstationary environments. In particular, each updated estimate of the state is computed from the previous estimate and the new input data, so only the previous estimate requires storage. In addition to eliminating the need for storing the entire past observed data, the Kalman filter is computationally more efficient than computing the estimate directly from all of those past data at each step of the filtering process.

Consider a linear, discrete time dynamical system described by the signal flow graph shown in Figure 4.1. In mathematical terms the signal flow graph embodies the following equations.

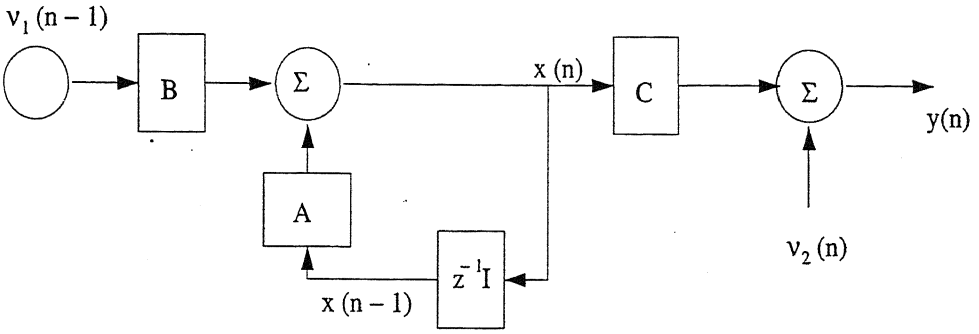


Figure 4.1: Block Diagram For Kalman Filter Algorithm

1. **Process Equation** :

$$\mathrm{x}(n) = \mathrm{A}\mathrm{x}(n-1) + \mathrm{B}\boldsymbol{\nu}_1(n-1) \tag{4.1}$$

Here $\mathrm{x} \in \mathcal{R}^M$ is the M-dimensional *state vector* that we are trying to estimate. $\boldsymbol{\nu}_1$ is a $M \times 1$ vector which represents the *process noise*. It is modeled as a zero mean white noise process whose correlation matrix is defined by :

$$E[\boldsymbol{\nu}_1(n)\boldsymbol{\nu}_1{}^H(k)] = \begin{cases} \mathrm{Q}_1(n) & n = k \\ \mathrm{O} & n \neq k \end{cases} \tag{4.2}$$

The process equation models an unknown physical stochastic phenomenon denoted by the state $\mathrm{x}(n)$ as the output of a linear dynamical system excited by the white noise $\boldsymbol{\nu}_1(n-1)$. The $M \times M$ *transition matrix* A relates the state at the previous time step $(n-1)$ to the state at the current time step n in the absence of any driving force or process noise. In practice it might change with each time step, but here it is assumed to be constant.

2. **Measurement Equation** :

$$\mathrm{y}(n) = \mathrm{C}\mathrm{x}(n) + \boldsymbol{\nu}_2(n) \tag{4.3}$$

Here $\mathrm{y} \in \mathcal{R}^N$ is the N-dimensional *observation vector*. It is actually the set of observed data that is used to estimate the vector x. C is a known $N \times M$ *measurement matrix*. The $N \times 1$ vector $\boldsymbol{\nu}_2$ is called *measurement noise*, modeled as a zero mean white noise process whose correlation matrix is :

$$E[\boldsymbol{\nu}_2(n)\boldsymbol{\nu}_2{}^H(k)] = \begin{cases} \mathbf{Q}_2(n) & n = k \\ \mathbf{O} & n \neq k \end{cases} \tag{4.4}$$

The measurement equation relates the observable output of the system $\mathbf{y}(n)$ to the state $\mathbf{x}(n)$.

It is assumed that $\mathbf{x}(0)$, the initial value of the state, is uncorrelated with both $\boldsymbol{\nu}_1(n)$ and $\boldsymbol{\nu}_2(n)$ for $n \geq 0$. The noise vectors $\boldsymbol{\nu}_1(n)$ and $\boldsymbol{\nu}_2(n)$ are statistically independent, so we may write

$$E[\boldsymbol{\nu}_1(n)\boldsymbol{\nu}_2{}^H(k)] = \mathbf{O} \qquad \forall n \text{ and } k \tag{4.5}$$

The Kalman filtering problem, namely, the problem of jointly solving the process and measurement equations for the unknown state in an optimal manner may be formally stated as :

Using the entire observed data, consisting of the observations $\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)$ *find for each* $n \geq 1$ *the minimum mean square estimate of the state* $\mathbf{x}(i)$.
The problem is called prediction if $i > n$.

4.2.1 Kalman Filter Algorithm

We define $\hat{\mathbf{x}}^-(n) \in \mathcal{R}^M$ to be the *a priori* state estimate at step n given knowledge of the process prior to step n and $\hat{\mathbf{x}}(n) \in \mathcal{R}^M$ to be the *a posteriori* state estimate at step n given observation vector $\mathbf{y}(n)$. We can then define *a priori estimate error* as

$$\mathbf{e}^-(n) = \mathbf{x}(n) - \hat{\mathbf{x}}^-(n) \tag{4.6}$$

and *a posteriori estimate error* as

$$\mathbf{e}(n) = \mathbf{x}(n) - \hat{\mathbf{x}}(n) \tag{4.7}$$

The *a priori* estimate error covariance is then

$$\mathbf{P}^-(n) = E[\mathbf{e}^-(n)(\mathbf{e}^-(n))^T] \tag{4.8}$$

and the *a posteriori* estimate error covariance is

$$\mathbf{P}(n) = E[\mathrm{e}(n)(\mathrm{e}(n))^T] \tag{4.9}$$

In deriving the equations for the Kalman filter, we begin with the goal of finding an equation that computes the *a posteriori* state estimate $\hat{\mathbf{x}}(n)$ as a linear combination of the *a priori* state estimate $\hat{\mathbf{x}}^-(n)$ and a weighted difference between the actual observation $\mathbf{y}(n)$ and the predicted observation $\mathbf{C}\hat{\mathbf{x}}^-(n)$ as given :

$$\hat{\mathbf{x}}(n) = \hat{\mathbf{x}}^-(n) + \mathbf{K}(n)(\mathbf{y}(n) - \mathbf{C}\hat{\mathbf{x}}^-(n)) \tag{4.10}$$

The difference $(\mathbf{y}(n) - \mathbf{C}\hat{\mathbf{x}}^-(n))$ is called the measurement *innovation*, or the *residual*. The residual reflects the discrepancy between the predicted observation and the actual observation. A residual of zero means that the two are in complete agreement. The $M \times N$ matrix $\mathbf{K}$ is known as the *Kalman Gain*. It is chosen such that the *a posteriori* error covariance is minimized.

The equations of the Kalman filter can be grouped into two groups : *time update* equations and *measurement update* equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. They are also known as *predictor* equations. The measurement update equations are responsible for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate. They are also called *corrector* equations.

Discrete Kalman Filter time update equations

$$\hat{\mathbf{x}}^-(n) = \mathbf{A}\hat{\mathbf{x}}(n-1) \tag{4.11}$$

$$\mathbf{P}^-(n) = \mathbf{A}\mathbf{P}(n-1)\mathbf{A}^T + \mathbf{B}\mathbf{Q}_1\mathbf{B}^T \tag{4.12}$$

Discrete Kalman Filter measurement update equations

$$\mathbf{K}(n) = \mathbf{P}^-(n)\mathbf{C}^T(\mathbf{C}\mathbf{P}^-(n)\mathbf{C}^T + \mathbf{Q}_2(n))^{-1} \tag{4.13}$$

$$\hat{\mathrm{x}}(n) = \hat{\mathrm{x}}^{-}(n) + \mathbf{K}(n)(\mathrm{y}(n) - \mathrm{C}\hat{\mathrm{x}}^{-}(n)) \tag{4.14}$$

$$\mathbf{P}(n) = (\mathbf{I} - \mathrm{K}(n)\mathrm{C})\mathbf{P}^{-}(n) \tag{4.15}$$

4.3 Dynamical Model

Dynamical models [5][11] characterize an object's behaviour over time. The prior model for the fitting problem has to be extended to deal with the problem of tracking the curve over a sequence of images. For tracking, it is necessary to provide not only a prior for the first frame, but also a prior for possible motions. It must have a deterministic part, giving the expected displacement between frames and a stochastic part to measure the uncertainty about the predictions of the model.

The dynamical model used is a second-order autoregressive process given by :

$$\mathrm{x}(n) = \mathbf{A}_1\mathrm{x}(n-1) + \mathbf{A}_2\mathrm{x}(n-2) + \mathbf{B}\boldsymbol{\nu}(n) \tag{4.16}$$

Here $\mathbf{x}$ is the shape space vector, $\boldsymbol{\nu}$ is the process noise. Matrices $\mathbf{A}_1$ and $\mathbf{A}_1$ represent the deterministic components of the dynamical model and matrix $\mathbf{B}$ represent the stochastic component of the dynamical model. We define an augmented vector $\mathbf{X}(n)$ to be :

$$\mathbf{X}(n) = \begin{pmatrix} \mathrm{x}(n) \\ \mathrm{x}(n-1) \end{pmatrix} \tag{4.17}$$

So the second order process can be written as :

$$\mathbf{X}(n) = \mathcal{A}\mathbf{X}(n-1) + \mathcal{B}\nu(n) \tag{4.18}$$

where

$$\mathcal{A} = \begin{pmatrix} \mathrm{A}_1 & \mathrm{A}_2 \\ \mathbf{I} & 0 \end{pmatrix} \qquad \mathcal{B} = \begin{pmatrix} \mathrm{B} \\ 0 \end{pmatrix} \tag{4.19}$$

For this thesis, this augmented model has been used for modeling the process and then Kalman Filter algorithm has been used for tracking.

Chapter 5

Fitting and Tracking in 3-Dimensions

5.1 Introduction

Fitting and tracking in 3-dimensional data is required, specially in the case of medical imaging like tracking the non-rigid movement of heart from 3-D MRI data. This chapter deals with the fitting and tracking of surfaces in 3-D. For analyzing a 3-dimensional volume, usually a suitable number of slices are taken depending upon the application and also the computational complexity as computation increases largely with the increase in the number of slices. So, for fitting in 3-dimensions, we have a certain number of slices of the volume.

5.2 Wavelet Shape Representation in 3-Dimensions

Let $\mathrm{s}(u) = (x(u), y(u), z(u))$ be the surface that represents the shape of the volume which we want to fit. The wavelet transform is applied independently to each of the $x(u)$, $y(u)$ and $z(u)$ functions. Then the surface can be represented in terms of the decomposition of $\mathrm{s}(u)$:

$$\mathrm{s}(u) = \sum_{l} \mathbf{c}_{j_0,l}\varphi_{j_0,l}(u) + \sum_{j=j_0}^{\infty}\sum_{l} \mathbf{d}_{j,l}\psi_{j,l}(u) \tag{5.1}$$

where

$$\mathbf{c}_{j_0,l} = (c_{j_0,l(x)}, c_{j_0,l(y)}, c_{j_0,l(z)})^T \tag{5.2}$$

$$\mathbf{d}_{j,l} = (d_{j,l(x)}, d_{j,l(y)}, d_{j,l(z)})^T \tag{5.3}$$

Here j_0 is an arbitrary starting scale. The $c_{j_0,l(x)}$ and $d_{j,l(x)}$ are the *scaling coefficients* and *wavelet coefficients* for the function $x(u)$, $c_{j_0,l(y)}$ and $d_{j,l(y)}$ are the *scaling coefficients* and *wavelet coefficients* for the function $y(u)$ and $c_{j_0,l(z)}$ and $d_{j,l(z)}$ are the *scaling coefficients* and *wavelet coefficients* for the function $z(u)$. $c_{j_0,l}$ and $d_{j,l}$ are the combined *scaling coefficients* and *wavelet coefficients* for the surface $\mathbf{s}(u)$.

If the surface is closed, we obtain the representation by taking the starting scale as 0 :

$$\mathbf{s}(u) = \mathbf{c}_{0,0}\varphi_{0,0}(u) + \sum_{j=0}^{\infty} \sum_{0 \leq l < 2^j} \mathbf{d}_{j,l}\psi_{j,l}(u) \tag{5.4}$$

The coefficients in the curve expansion is defined as :

$$\mathbf{c}_{0,0} = (\langle x(u), \varphi_{0,0}(u)\rangle, \langle y(u), \varphi_{0,0}(u)\rangle, \langle z(u), \varphi_{0,0}(u)\rangle)^T \tag{5.5}$$

$$\mathbf{d}_{j,l} = (\langle x(u), \psi_{j,l}(u)\rangle, \langle y(u), \psi_{j,l}(u)\rangle, \langle z(u), \psi_{j,l}(u)\rangle)^T \tag{5.6}$$

Here the scaling and wavelet functions are normalized. Considering only a finite number of coefficients, which is usually done in practice, we get the representation as :

$$\mathbf{s}(u) = \mathbf{c}_{0,0}\varphi_{0,0}(u) + \sum_{j=0}^{J} \sum_{0 \leq l < 2^j} \mathbf{d}_{j,l}\psi_{j,l}(u) \tag{5.7}$$

This surface expansion can be written in concise form. First the scaling and wavelet functions are expressed in vector form as :

$$\mathbf{G}_W(u) = (\varphi_{0,0}(u), \psi_{0,0}(u), \ldots, \psi_{J-1,2^{J-1}-1}(u))^T \tag{5.8}$$

then the coordinate functions can be written as :

$$x(u) = \mathbf{G}_W(u)^T\mathbf{w}^x \quad \text{and} \quad y(u) = \mathbf{G}_W(u)^T\mathbf{w}^y \quad z(u) = \mathbf{G}_W(u)^T\mathbf{w}^z \tag{5.9}$$

with scaling and wavelet coefficient vectors $\mathbf{w}^x$, $\mathbf{w}^y$ and $\mathbf{w}^z$ defined as :

$$\mathbf{w}^x = (c_{0,0(x)}, d_{0,0(x)}, \ldots, d_{J-1,2^{J-1}-1(x)})^T \tag{5.10}$$

$$\mathbf{w}^y = \left(c_{0,0(y)}, d_{0,0(y)}, \ldots, d_{J-1,2^{J-1}-1(y)}\right)^T \tag{5.11}$$

$$\mathbf{w}^z = \left(c_{0,0(z)}, d_{0,0(z)}, \ldots, d_{J-1,2^{J-1}-1(z)}\right)^T \tag{5.12}$$

To describe the curve in matrix form, we define :

$$\mathbf{F}_W(u) = \mathbf{I}_3 \otimes \mathbf{G}_W(u)^T = \begin{pmatrix} \mathbf{G}_W(u)^T & \mathbf{0}^T{}_N & \mathbf{0}^T{}_N \\ \mathbf{0}^T{}_N & \mathbf{G}_W(u)^T & \mathbf{0}^T{}_N \\ \mathbf{0}^T{}_N & \mathbf{0}^T{}_N & \mathbf{G}_W(u)^T \end{pmatrix} \tag{5.13}$$

where $\otimes$ denotes Kronecker product and $\mathbf{0}_N$ a null vector of N components where $N = 2^J$.

If we define $\mathbf{w} = \left(\mathbf{w}^{xT}, \mathbf{w}^{yT}, \mathbf{w}^{zT}\right)^T$, then we may write

$$\mathbf{s}(u) = \mathbf{F}_W(u)\mathbf{w} \tag{5.14}$$

5.3 Wavelet Probabilistic Deformation Model in 3-Dimensions

Like in 2-D case, in 3-D also, the fitting and tracking problem can be formulated in probabilistic terms that allow an explicit representation for the uncertainty. For formulation of the wavelet probabilistic model, we assume that the coefficients are independent. Under the independence assumption, modeling reduces to simply specifying the marginal distribution of each scaling and wavelet coefficients. In this thesis, the wavelet coefficients are assumed to be Gaussian distributed.

$$\mathbf{d}_{j,k} = \left(d_{j,k(x)}, d_{j,k(y)}, d_{j,k(z)}\right)^T \sim N(\mathbf{0}_3, \sigma^2{}_j\mathbf{I}_3) \tag{5.15}$$

$$\sigma_j = 2^{-j\beta}\sigma_{DEF}, \qquad \beta > 0 \qquad \text{and} \qquad \sigma_{DEF} > 0 \tag{5.16}$$

The distribution of the scaling coefficient $\mathbf{c}_{0,0}$ is also taken as Gaussian and it is taken to be independent of the non-translation components $\mathbf{d}_{j,k}$.

$$\mathbf{c}_{0,0} = \begin{pmatrix} c_{0,0(x)} \\ c_{0,0(y)} \\ c_{0,0(z)} \end{pmatrix} \sim N(\mathbf{0}_3, \sigma^2{}_{TR}\mathbf{I}_3) \tag{5.17}$$

Assuming these distributions, the wavelet probabilistic model can be expressed in matrix form as

$$P(\mathbf{r}) \equiv P(\mathbf{w}) \propto exp(-\frac{1}{2}\mathbf{w}^T \mathrm{S}_w \mathbf{w}), \qquad \mathbf{r} = \mathbf{F}_W \mathbf{w} \tag{5.18}$$

with

$$\Sigma = \mathrm{S}_w{}^{-1} = \mathrm{I}_3 \otimes \begin{pmatrix} \sigma^2{}_{TR} & 0 & 0 & \dots & 0 \\ 0 & \sigma^2{}_{DEF}2^{-2\beta(0)} & 0 & \dots & 0 \\ 0 & 0 & \sigma^2{}_{DEF}2^{-2\beta(1)} & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & \sigma^2{}_{DEF}2^{-2\beta(J-1)} \end{pmatrix} \tag{5.19}$$

Here β is related to the smoothness of the deformation and $\sigma^2{}_{TR}$ and $\sigma^2{}_{DEF}$ are related to the uncertainty of the contour.

5.4 Shape Space for 3-Dimensions

For surface fitting in the 3-dimensional case, as already mentioned in the previous sections, the x, y and z functions are found out from the given slices of the given volume and the wavelet transform is applied separately to these three functions. Let the combined wavelet vector consisting of the scaling and wavelet coefficients for all the 3-dimensions be denoted as $\mathbf{w}$. i.e.

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}^x \\ \mathbf{w}^y \\ \mathbf{w}^z \end{pmatrix} \tag{5.20}$$

A linear shape apace $\mathcal{M} = \mathcal{L}(H, \bar{\mathbf{w}})$ is a linear mapping of a *shape space vector* $\mathrm{x} \in \mathbb{R}^{N_\mathrm{X}}$ to the vector $\mathbf{w} \in \mathbb{R}^{N_W}$:

$$\mathbf{w} = H\mathrm{x} + \bar{\mathbf{w}} \tag{5.21}$$

Here H is known as the *shape matrix*. $\bar{\mathbf{w}}$ represents a base or template surface against which shape variations are measured.

In this thesis, the concept of Planar Affine Shape in 2-dimensions has been extended for the formulation of the shape space matrix and shape space vector for the 3-D case. Let $\mathrm{s}_0(u)$ be the template surface. This shape space can be viewed as a class of all linear transformations that can be applied to $\mathrm{s}_0(u)$:

$$\mathrm{s}(u) = \mathbf{t} + M\mathrm{s}_0(u) \tag{5.22}$$

where $\mathbf{t} = (t_1, t_2, t_3)^T$ is a three-dimensional translational vector and M is a 3×3 matrix, so that $\mathbf{t}$ and M together represent the 12 degrees of freedom of the space. This class can be represented as a shape space with template $\bar{\mathrm{w}}$ and shape matrix

$$H = \begin{pmatrix} \mathrm{T} & 0 & 0 & \bar{\mathrm{w}}^x & 0 & 0 & 0 & 0 & 0 & 0 & \bar{\mathrm{w}}^y & \bar{\mathrm{w}}^z \\ 0 & \mathrm{T} & 0 & 0 & \bar{\mathrm{w}}^y & 0 & 0 & 0 & \bar{\mathrm{w}}^x & \bar{\mathrm{w}}^z & 0 & 0 \\ 0 & 0 & \mathrm{T} & 0 & 0 & \bar{\mathrm{w}}^z & \bar{\mathrm{w}}^x & \bar{\mathrm{w}}^y & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.23}$$

where $\mathbf{T} = (10\ldots0)^T$ and $\mathbf{0} = (00\ldots0)^T$ are vectors each with N components. The first three columns of H represents translations in the 3-dimensions. Linear combinations of the other columns represents the remaining motions.

With this choice of the shape space matrix, the shape space vector is :

$$\mathrm{x} = (t_1, t_2, t_3, M_{11} - 1, M_{22} - 1, M_{33} - 1, M_{31}, M_{32}, M_{21}, M_{23}, M_{12}, M_{13})^T \tag{5.24}$$

5.5 Probabilistic Model In Shape Spaces

The probabilistic model in wavelet space induces a probabilistic model in shape space:

$$P(\mathrm{x}) \propto exp(-\frac{1}{2}\mathrm{x}^T\mathrm{S}_x\mathrm{x}) \tag{5.25}$$

where S_x is the covariance matrix of the vector x. Since $\mathrm{w} = H\mathrm{x} + \bar{\mathrm{w}}$, so we can write

$$\mathrm{x} = H^+(\mathrm{w} - \bar{\mathrm{w}}) \tag{5.26}$$

where H^+ is the pseudo-inverse of H. Here since $\bar{\mathrm{w}}$ is the combined wavelet vector of the base surface, say $\bar{\mathrm{x}}$, so $\bar{\mathrm{x}} = 0$. This is also evident from the fact that shape space vectors are basically deviations from the base or template surface. Then the inverse of the covariance matrix S_x is given by :

$$\mathrm{S}_x{}^{-1} = H^+\mathrm{S}_w{}^{-1}H^{+T} \tag{5.27}$$

5.6 The Fitting Problem in 3-Dimensions

The fitting problem can be defined in Bayesian terms as follows :
The prior density is given by the probabilistic model in shape space, i.e.

$$P(\mathrm{x}) \propto exp(-\frac{1}{2}\mathrm{x}^T\mathrm{S}_x\mathrm{x}) \tag{5.28}$$

From the image, some measurements of the contour x_D are obtained. The likelihood function is defined as :

$$P(\mathbf{r}_D \mid \mathbf{r}) = P(\mathbf{w}_D \mid \mathrm{w}) = P(\mathrm{x}_D \mid \mathrm{x}) \propto exp(-\frac{1}{2}(\mathrm{x} - \mathrm{x}_D)^T\mathrm{S}_D(\mathrm{x} - \mathrm{x}_D)) \tag{5.29}$$

with information matrix S_D given by :

$$\mathrm{S}_D{}^{-1} = H^+\sigma_D{}^2H^{+T} \tag{5.30}$$

Here σ_D is the variance of $\mathbf{w}_D - \bar{\mathbf{w}}$ where the wavelet and shape spaces are related by :

$$\mathbf{r}_D = \mathbf{F}_W\mathbf{w}_D \qquad \mathbf{w}_D = H\mathrm{x}_D + \bar{\mathbf{w}} \tag{5.31}$$

From Bayesian estimation, we get the maximum aposteriori estimate $\hat{\mathrm{x}}$ as:

$$\hat{\mathrm{x}} = (\mathrm{S}_x + \mathrm{S}_D)^{-1}\mathrm{S}_D\mathrm{x}_D \tag{5.32}$$

where the measurement vector x_D is given by $\mathrm{x}_D = H^+(\mathbf{w}_D -- \bar{\mathbf{w}})$.

As in the 2-dimensional case, in 3-dimension also, tracking has been implemented using Kalman Filter algorithm.

Chapter 6

Implementation and Results

6.1 Introduction

In this chapter, the algorithms for 2-D and 3-D fitting and tracking are given along with the results. The implementation has been done in *C* and the images used were of Bit Map Image format (.bmp). Some of the images used are real images while some are synthetic. The 3-D visualization has been done using OpenGL.

6.2 Algorithm for 2-D Curve fitting and Tracking

1. The image sequence is read i.e. intensity values of all the pixels of all the images are stored.

2. A hand drawn prior is provided around the object of interest in the first image.

3. The x and y co-ordinates of the points on the hand drawn prior are found out and stored.

4. The wavelet decomposition (wavelet series expansion) is performed on the above x-function and y-function separately i.e. the wavelet coefficient vectors $\bar{\mathrm{w}}^x$ and $\bar{\mathrm{w}}^y$ are computed. Here Haar wavelets have been used for the implementation. Then the combined wavelet vector is formed, which is given

by

$$\bar{\mathbf{w}} = \begin{pmatrix} \bar{\mathbf{w}}^x \\ \bar{\mathbf{w}}^y \end{pmatrix} \tag{6.1}$$

5. The shape space matrix H is formed using these $\bar{\mathbf{w}}^x$ and $\bar{\mathbf{w}}^y$ values for the 2-dimensional case. Here Planar Affine shape space has been used. Also the pseudo-inverse of H, i.e. H^+ is found out.

6. From the hand drawn prior, the edge of the object of interest in the first image is measured. For each point on the prior, radially inwards and radially outwards a certain number of pixels (say 15 or 20) are chosen and their intensity values are stored. Then the gradient information is used to detect the probable edge of the object since at the edge there will be a significant change in the intensity values. The x and y co-ordinates of these measured points are stored.

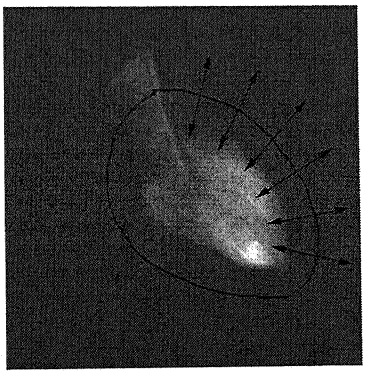


Figure 6.1: Hand drawn prior around the heart. Arrows show the direction of searching from the hand drawn prior

7. The wavelet coefficients i.e. $\mathbf{w}_D{}^x$ and $\mathbf{w}_D{}^y$ of the measured x and y functions are calculated. Here also Haar wavelets have been used for the wavelet decomposition. Then the combined measured wavelet vector is computed.

$$\mathbf{w}_D = \begin{pmatrix} \mathbf{w}_D{}^x \\ \mathbf{w}_D{}^y \end{pmatrix}$$

8. The variance ${\sigma_D}^2$ of $\mathbf{w}_D - \bar{\mathbf{w}}$ is computed.

9. The covariance matrix S_D of $\mathbf{x}_D$ is computed where ${\mathrm{S}_D}^{-1}$ is given by :

$$ {\mathrm{S}_D}^{-1} = H^{+}{\sigma_D}^{2}H^{+T} \qquad (6.3) $$

10. The covariance matrix S_x of $\mathbf{x}$ is calculated as:

$$ {\mathrm{S}_x}^{-1} = H^{+}{\mathrm{S}_w}^{-1}H^{+T} \qquad (6.4) $$

11. The shape space vector $\mathbf{x}_D$ for the measured wavelet vector $\mathbf{w}_D$ is computed using the following relation:

$$ \mathbf{x}_D = H^{+}(\mathbf{w}_D - \bar{\mathbf{w}}) \qquad (6.5) $$

12. The value of the variable *image number* is set to 1 (i.e. for the first image)

13. If *image number* = 1, the maximum aposteriori estimate of the fitted shape space vector $\hat{\mathbf{x}}$ is :

$$ \hat{\mathbf{x}} = (\mathrm{S}_x + \mathrm{S}_D)^{-1}\mathrm{S}_D\mathbf{x}_D \qquad (6.6) $$

else the aposteriori estimate of the fitted shape space vector is computed from the Kalman Filter algorithm.

14. The combined wavelet vector $\hat{\mathbf{w}}$ corresponding to the fitted shape space vector is computed using the following relation :

$$ \hat{\mathbf{w}} = H\hat{\mathbf{x}} + \bar{\mathbf{w}} \qquad \text{where} \qquad \hat{\mathbf{w}} = \begin{pmatrix} \hat{\mathbf{w}}^{x} \\ \hat{\mathbf{w}}^{y} \end{pmatrix} \qquad (6.7) $$

from which $\hat{\mathbf{w}}^{x}$ and $\hat{\mathbf{w}}^{y}$ are found out.

15. The x and y functions of the fitted curve is constructed by taking the inverse wavelet transform of $\hat{\mathbf{w}}^{x}$ and $\hat{\mathbf{w}}^{y}$ respectively.

16. The value of *image number* is incremented by 1. If *image number* is less than or equal to the total number of images, then the algorithm returns to step 13.

6.3 Algorithm for 3-D Curve fitting and Tracking

1. The image sequence (all the slices of all the images) is read i.e. the intensity values of all the pixels of all the images are stored.

2. Hand drawn priors are provided around the object of interest for all the slices of the first image.

3. The x, y and z co-ordinates of the points on all the hand drawn priors are found out and stored.

4. The x co-ordinates of all the slices of the first image are arranged sequentially in a single array. The same procedure is followed for the y and z co-ordinates.

5. The wavelet decomposition (wavelet series expansion) is performed on the above x-function, y-function and z-function calculated in the previous step separately i.e. the wavelet coefficient vectors $\bar{\mathbf{w}}^x$, $\bar{\mathbf{w}}^y$ and $\bar{\mathbf{w}}^z$ are computed. Here Haar wavelets have been used for the implementation. Then the combined wavelet vector is formed as follows

$$\bar{\mathbf{w}} = \begin{pmatrix} \bar{\mathbf{w}}^x \\ \bar{\mathbf{w}}^y \\ \bar{\mathbf{w}}^z \end{pmatrix} \tag{6.8}$$

6. The shape space matrix H is formed by taking these $\bar{\mathbf{w}}^x$, $\bar{\mathbf{w}}^y$ and $\bar{\mathbf{w}}^z$ values. Here the 3-dimensional extension of the Planar Affine shape space has been used. Also the pseudo-inverse of H, i.e. H^+ is computed.

7. From the hand drawn priors, the edges of the object of interest for all the slices in the first image are measured. For each point on the priors, radially inwards and radially outwards a certain number of pixels (say 15 or 20) are chosen and their intensity values are stored. Then the gradient information is used to detect the probable edges of the object since at the edge there will be a significant change in the intensity values. The x, y and z co-ordinates of these measured points are stored.

8. The wavelet coefficients i.e. $\mathbf{w_D}^x$, $\mathbf{w_D}^y$ and $\mathbf{w_D}^z$ of the measured x, y and z functions are calculated. Here also Haar wavelets have been used for the wavelet decomposition. Then the combined measured wavelet vector is computed.

$$\mathbf{w}_D = \begin{pmatrix} \mathbf{w_D}^x \\ \mathbf{w_D}^y \\ \mathbf{w_D}^z \end{pmatrix} \tag{6.9}$$

9. The variance ${\sigma_D}^2$ of $\mathbf{w}_D - \bar{\mathbf{w}}$ is computed.

10. The covariance matrix S_D is computed where ${\mathrm{S}_D}^{-1}$ is given by:

$${\mathrm{S}_D}^{-1} = H^+ {\sigma_D}^2 H^{+T} \tag{6.10}$$

11. The covariance matrix S_x of x is calculated as:

$${\mathrm{S}_x}^{-1} = H^+ {\mathrm{S}_w}^{-1} H^{+T} \tag{6.11}$$

12. The shape space vector $\mathbf{x}_D$ for the measured wavelet vector $\mathbf{w}_D$ is computed using the following relation:

$$\mathbf{x}_D = H^+(\mathbf{w}_D - \bar{\mathbf{w}}) \tag{6.12}$$

13. The value of the variable *image number* is set to 1 (i.e. for the first image)

14. If *image number* = 1, the maximum aposteriori estimate of the fitted shape space vector $\hat{\mathbf{x}}$ is:

$$\hat{\mathbf{x}} = (\mathrm{S}_x + \mathrm{S}_D)^{-1} \mathrm{S}_D \mathbf{x}_D \tag{6.13}$$

else the aposteriori estimate of the fitted shape space vector is computed from the Kalman Filter algorithm.

15. The combined wavelet vector $\hat{\mathbf{w}}$ corresponding to the fitted shape space vector is computed using the following relation :

$$\hat{\mathbf{w}} = H\hat{\mathbf{x}} + \bar{\mathbf{w}} \qquad \text{where} \qquad \hat{\mathbf{w}} = \begin{pmatrix} \hat{\mathbf{w}}^x \\ \hat{\mathbf{w}}^y \\ \hat{\mathbf{w}}^z \end{pmatrix} \tag{6.14}$$

from which $\hat{\mathbf{w}}^x$, $\hat{\mathbf{w}}^y$ and $\hat{\mathbf{w}}^y$ are found out.

16. The reconstructed x, y and z functions is computed by taking the inverse wavelet transform of $\hat{\mathbf{w}}^x$, $\hat{\mathbf{w}}^y$ and $\hat{\mathbf{w}}^z$ respectively. This is actually the coordinates of all the slices of the image. So the x, y and z functions of the different slices are then separated out.

17. The value of *image number* is incremented by 1. If *image number* is less than or equal to the total number of images, then the algorithm returns to step 14.

6.4 Simulation Results For 2-D Fitting

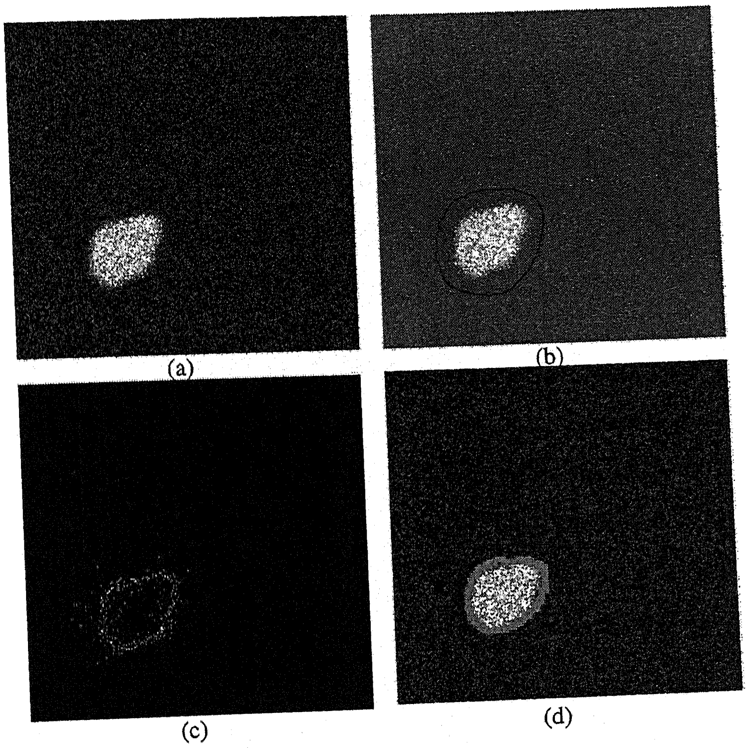


Figure 6.2: (a) Noisy synthetic image, (b) Hand drawn prior provided around the object of interest, (c) Measurement data obtained from the noisy image, (d) Fitted curve around the object of interest

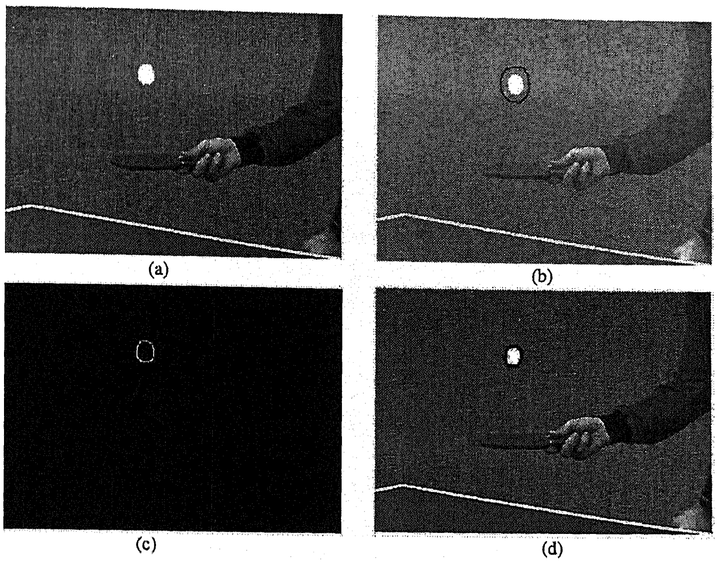


Figure 6.3: (a) Real image of a table tennis ball, (b) Hand drawn prior provided around the ball, (c) Measurement data (d) Fitted curve

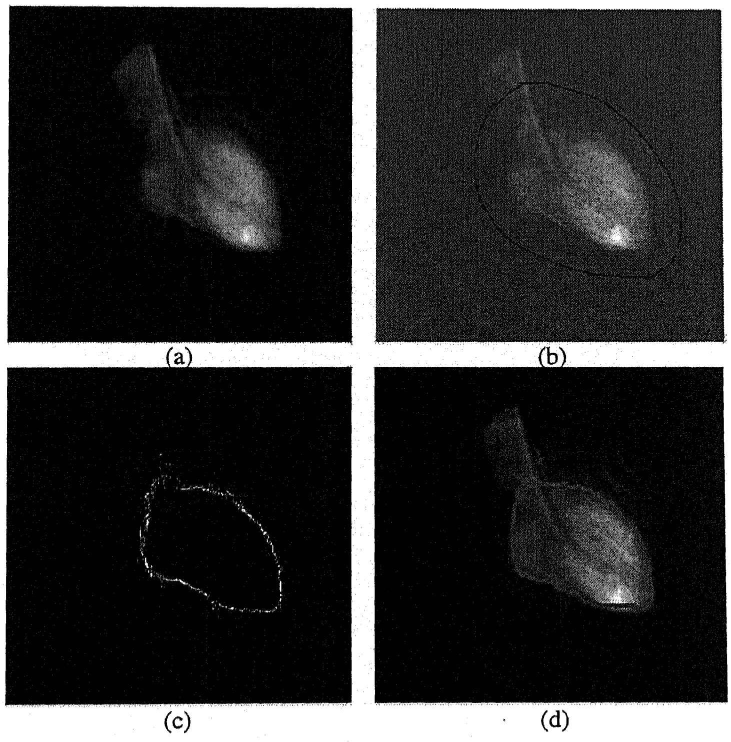


Figure 6.4: (a) Image of a human heart, (b) Hand drawn prior provided around the heart, (c) Measurement data (d) Fitted curve

6.5 Simulation Results For 2-D Tracking

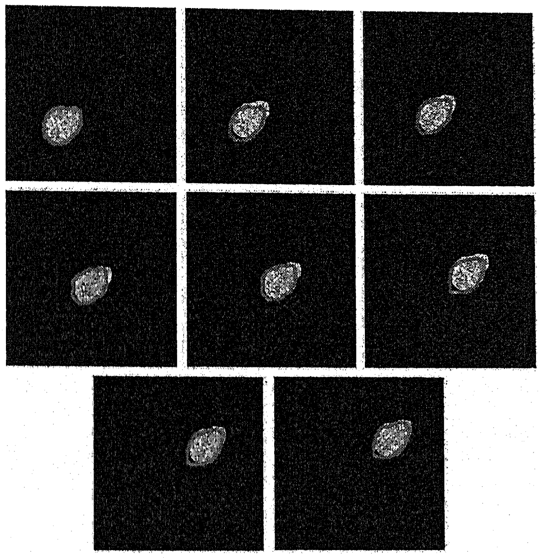


Figure 6.5: Rowwise from top : Some Frames from the tracking results in a noisy synthetic image

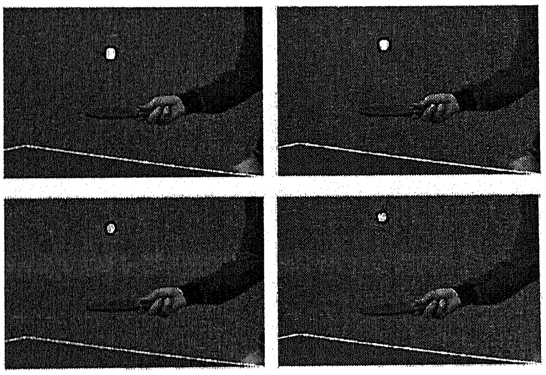


Figure 6.6: Rowwise from top : Some Frames from the tracking results of a table tennis ball in a real image

6.6 Simulation Results For 3-D Fitting

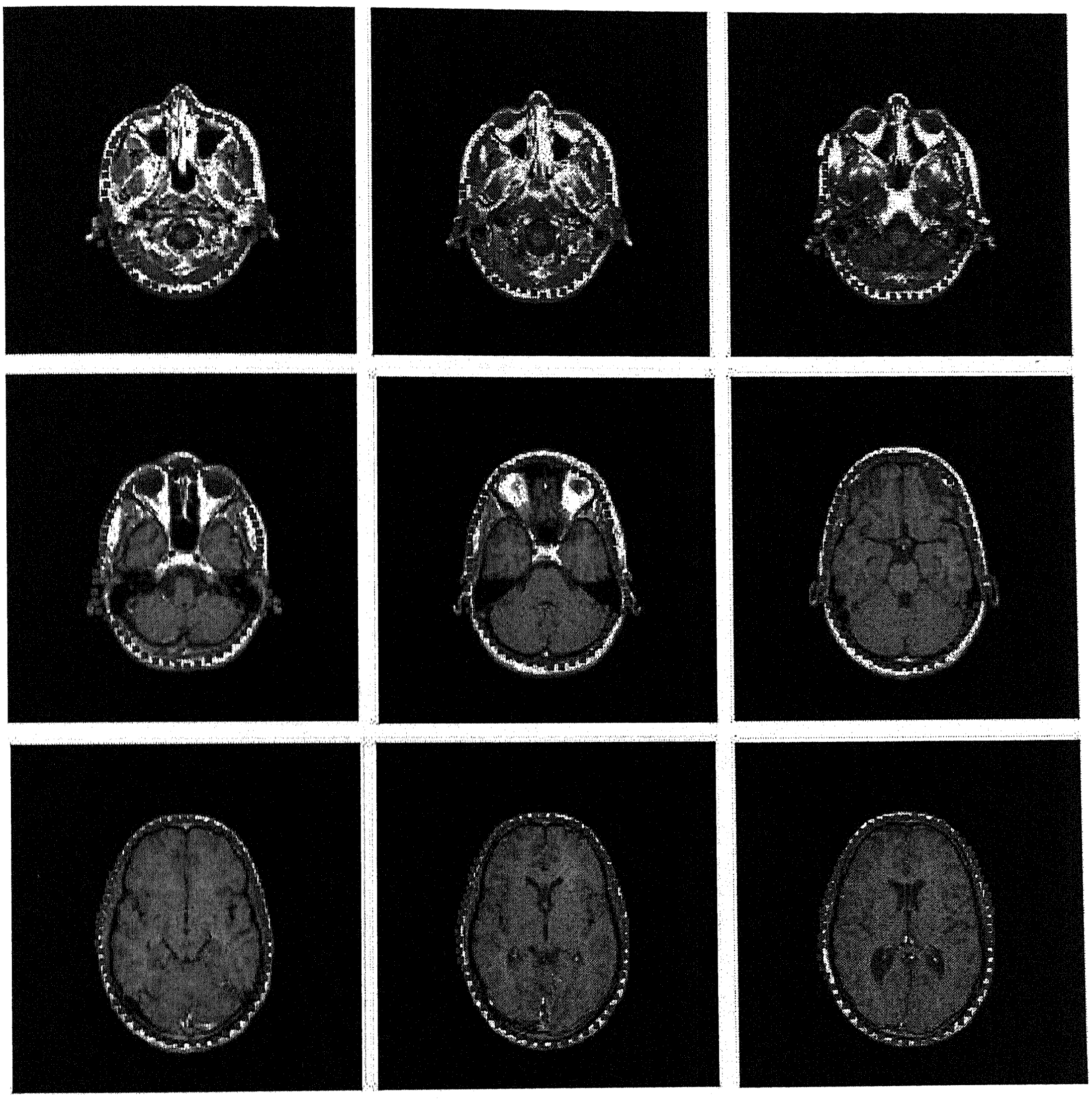


Figure 6.7: 3-D fitting results on MRI slices of human brain

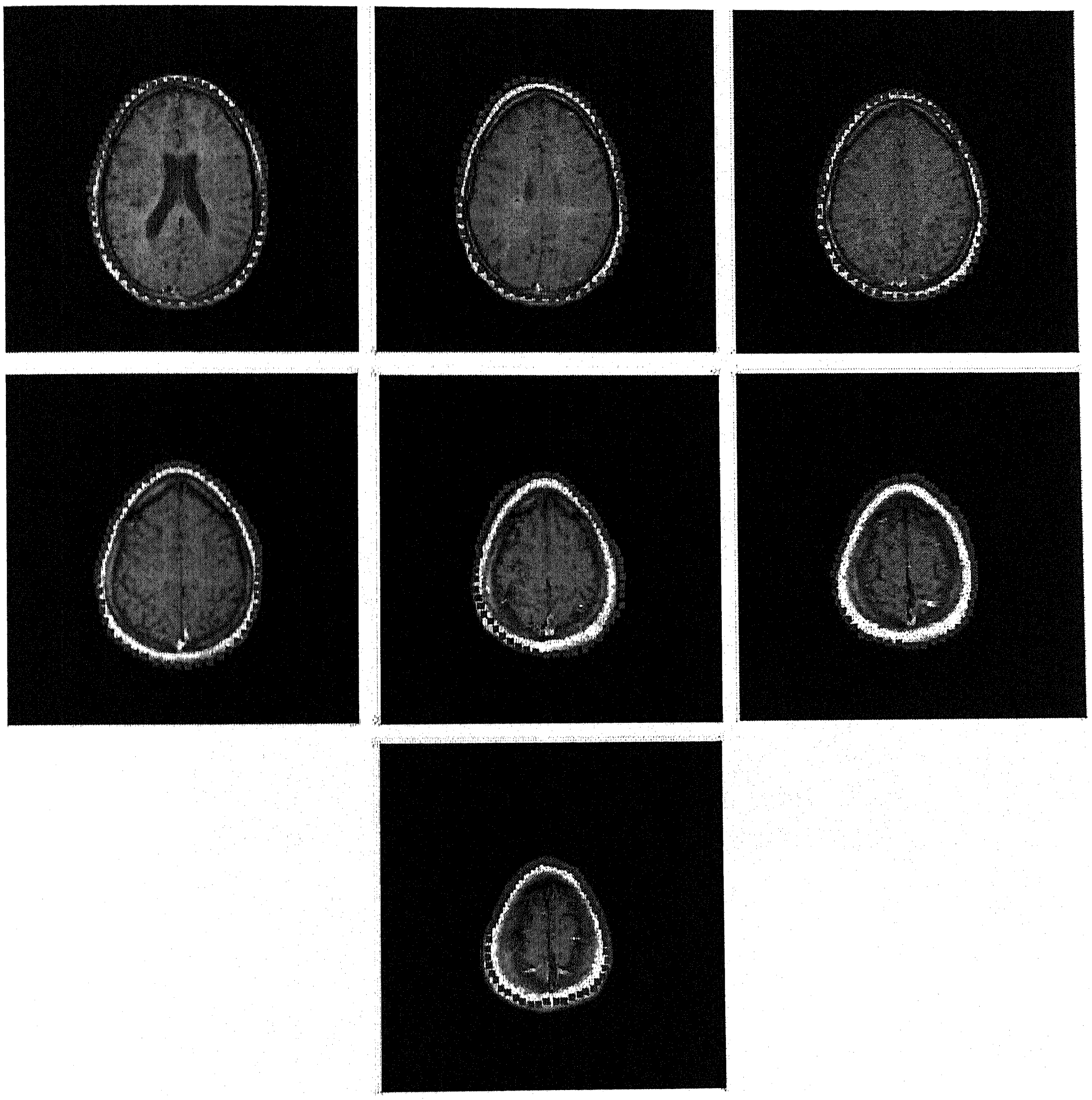


Figure 6.8: 3-D fitting results on MRI slices of human brain: contd.

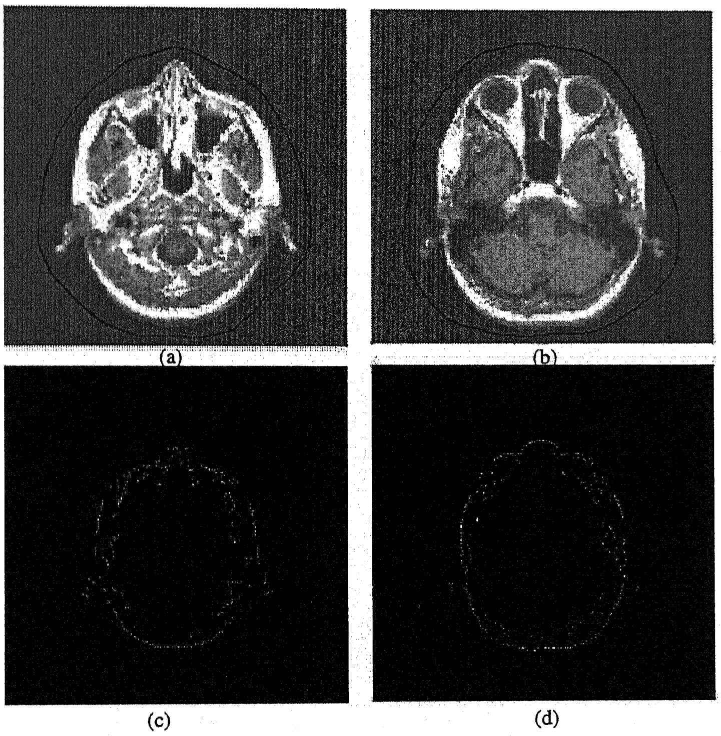


Figure 6.9: (a) Hand drawn prior for slice 1, (b) Hand drawn prior for slice 5, (c) Measurement for slice 1, (d) Measurement for slice 5

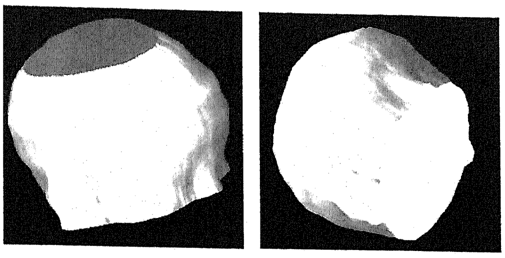
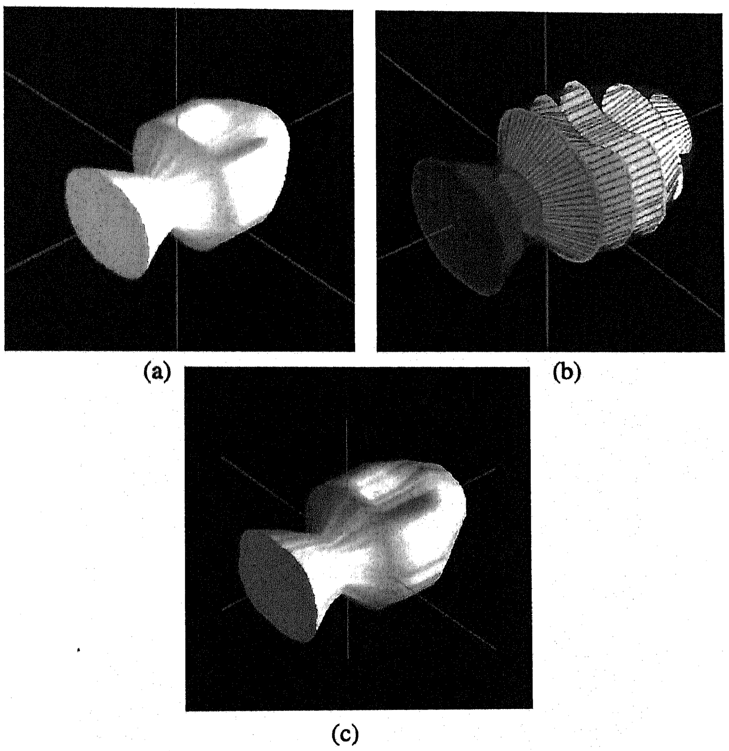


Figure 6.10: 3D rendering for the fitted MRI data



(a)

(b)

(c)

Figure 6.11: (a) 3D rendering of synthetic 3-D data, (b) Slice data for the above image with the fitted data shown as wire frame, (c) 3D rendering of fitted data

6.7 Simulation Results For 3-D Tracking

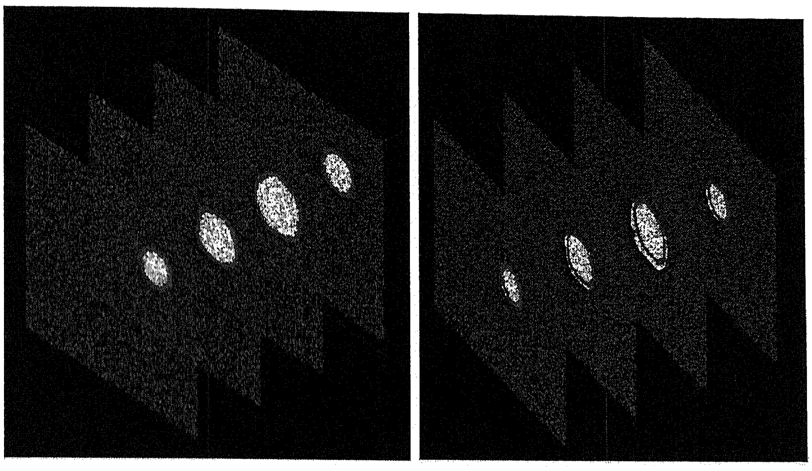


Figure 6.12: 2 frames from the 3D synthetic image used for tracking

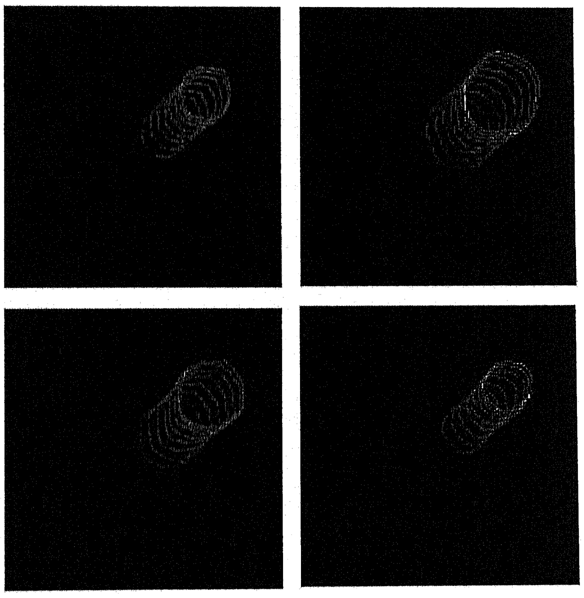


Figure 6.13: Tracked movement of the individual slices

Chapter 7

Conclusion and Future Scope

The thesis concentrates on fitting and tracking algorithms in 2-dimensions and 3-dimensions using wavelets. The inter-relation between wavelets and function spaces namely Sobolev space has been utilized to enforce a degree of smoothness to the fitted curves. The contour fitting has been done using Bayesian estimation. In the thesis Haar wavelets have been used for the wavelet decomposition of the image edges. The algorithms developed for both 2-D and 3-D have shown satisfactory results. Specially in the 3-D case the fitting has been done on slice data and also the 3-D rendering has been done using *OpenGL* for proper visualization purpose. The rendering algorithm also shows quite good results. This fitting algorithm is the initial step for developing the tracking algorithm. Tracking has been performed for both 2-dimensional and 3-dimensional data. For the tracking purpose Kalman filter has been used. The tracking results are also satisfactory as is evident from the examples of the previous chapter. As the 3-D visualization of 3-D tracking is quite complicated only the movement of the individual slices have been shown. The results make it evident that this framework which combines the wavelets and the function spaces can be used for both tracking and fitting purposes successfully. Some of the practical applications can be in medical imaging or tracking of an object in real images. In recent times use of 3-D data for medical imaging is quite popular. So this algorithm can be definitely be used for analyzing such data, like the movements of the human heart can be tracked using this algorithm.

7.1 Future Scope

Some of the directions of future research may include:

- In this research work Haar wavelets have been used for implementation. However proper choice of wavelets is very important for proper fitting and tracking, depending on the application. Hence the algorithm should be tested using other wavelets also.

- For tracking a proper dynamic model is required for successful tracking of the object of interest. The dynamic model should reflect the possible movements of the object. Some frames showing the moving object can be used for extracting this information and can be incorporated in the tracking algorithm.

- Simple Kalman filter has been used for implementing the tracking algorithm. But for images containing more clutter this algorithm may not be a robust one. For more robust and accurate results *extended Kalman filter* or *condensation algorithm* can be used.

- Here planar affine shape space has been used for representing the deformation of the object. However other shape spaces which can represent the deformation in a better way can be used.

References

[1] T. McInerney and D. Terzopoulos, Deformable models in medical images analysis: a survey, *Med. Image Anal.* 1 (1996) pp. 91 108.

[2] M. Kass, A. Witkin and D. Terzopoulos, Snakes: active contour models, *Int. J. Comput. Vision* 1, (1998), pp. 321-331.

[3] P. Saint-Marc, H. Rom and G. Medioni, B-Spline Contour Representation and Symmetry Detection, *IEEE Transactions on Pattern Analysis. and Machine Intelligence*, Vol. 15, pp. 1191-1197, November 1993.

[4] L. H. Staib and J. S. Duncan, Boundary Finding with Parametrically Deformable Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, pp. 1061-1075, November 1992.

[5] F. P. Nava and A. F. Martel, Wavelet modeling of contour deformations in Sobolev spaces for fitting and tracking applications, *Pattern Recognition* 36 (2003), pp. 1119-1130.

[6] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Second Edition, 2003, Pearson Education.

[7] N. J. Fliege, *Multirate Digital Signal Processing*, 1994, John Wiley and Sons.

[8] S. G. Mallat, A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 674-693, July 1989.

[9] E. Hernández and G. Weiss, *A First Course on Wavelets*, CRC Press, New York.

[10] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, Third Edition, 1991, McGraw-Hill Inc.

[11] M. A. Isard, *Visual Motion Analysis by Probabilistic Propagation of Conditional Density*, PhD thesis, 1998, University of Oxford.

[12] M. A. Isard, A. Blake and D. Reynard, *Learning to track the visual motion of contours*, University of Oxford.

[13] Simon Haykin, *Adaptive Filter Theory*, Fourth Edition, Pearson Education.

[14] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, University of North Carolina, 2003.